REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
20-04-2017	Final Report	1-Feb-2015 - 31-Aug-2016

4. TITLE AND SUBTITLE
Final Report: High Performance Techniques to Identify Source of Digital Images Using Multimedia Forensics

5a. CONTRACT NUMBER
W911NF-15-2-0013

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
Matthew C. Stamm

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Drexel University
Office of Research
3201 Arch Street, Suite 100
Philadelphia, PA 19104 -2875

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
66439-LS-FRD.3

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This report details the work performed on the project "High-Performance Techniques to Identify the Source of Digital Images Using Multimedia Forensics".

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Matthew Stamm
UU	UU	UU	UU		19b. TELEPHONE NUMBER 215-895-5894

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

Final Report: High Performance Techniques to Identify Source of Digital Images Using Multimedia Forensics

ABSTRACT

This report details the work performed on the project "High-Performance Techniques to Identify the Source of Digital Images Using Multimedia Forensics".

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
04/20/2017	1	Chen Chen, Matthew Stamm. Camera model identification framework using an ensemble of demosaicing features, 2015 IEEE International Workshop on Information Forensics and Security (WIFS). 16-NOV-15, Roma, Italy. : ,
04/20/2017	2	Xinwei Zhao, Matthew Stamm. Computationally efficient demosaicing filter estimation for forensic camera model identification, 2016 IEEE International Conference on Image Processing (ICIP). 25-SEP-16, Phoenix, AZ, USA. : ,

TOTAL: **2**

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received Paper

TOTAL:

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

TOTAL:

Number of Manuscripts:

Books

Received Book

TOTAL:

Received Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	DISCIPLINE
Xinwei Zhao	100	Electrical Engineering
Owen Mayer	100	Electrical Engineering
Chen Chen	100	Electrical Engineering
Belhassen Bayar	100	Electrical Engineering
Leland Machen	100	Applied Mathematics
FTE Equivalent:	**5.00**	
Total Number:	**5**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Matthew C. Stamm	0.06	
Nagarajan Kandasamy	0.06	
FTE Equivalent:	**0.12**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME	
Leland Machen	
Total Number:	**1**

Names of personnel receiving PHDs

NAME
Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See Attachement

Technology Transfer

FINAL REPORT

Submitted to:
The Army Research Office (ARO)
Defense Forensics and Biometrics Agency (DFBA)

Agreement Administered By:
Office of Naval Research Regional Office (Chicago)

Federal Grant Number
W911NF-15-2-0013

Project Title
High-Performance Techniques to Identify the Source of Digital Images Using Multimedia Forensics

Prepared & Submitted By

Dr. Matthew C. Stamm, Project PI
Elec. & Comp. Eng. Dept., Drexel University
3141 Chestnut Street
Philadelphia, PA 19104
Phone: (215) 895 5894
Email: mstamm@coe.drexel.edu

DUNS and EIN Numbers
DUNS Number: 002604817
EIN Number: 23-1352630

Recipient Organization
Drexel University
3201 Arch Street, Suite 100
Philadelphia, PA 19104

Project Period
01 February 2015–31 August 2016

End Date of Reporting Period
31 August 2016

Report Term
Final Technical Report

Signature of Submitting Official

Contents

1 Accomplishments

1.1 Project Goals

This project addressed the following tasks, aimed at developing technologies to identify the source of digital images using multimedia forensics:

- **Task 1.** Implementing forensic algorithms that identify traces specific to the make and model of the device that captured an image.
- **Task 2.** Developing data fusion algorithms that will optimally combine these forensic traces and use them to identify the make and model of an image's source device.
- **Task 3.** Creating a high-performance and scalable implementation of our algorithms through the design of highly parallel algorithms suitable for use on multi-core CPUs and graphics processing units (GPUs).
- **Task 4.** Developing techniques to enable large-scale crowdsourced training of our algorithms and its associated software implementation.

Milestones and Deliverables

This project will provided major deliverables in six-month increments over a period of 18 months. These deliverables and their associated schedule of delivery are listed below. In addition to these deliverables, monthly execution reports to OUSD (AT&L) were delivered, as well as a final report (i.e. this report) upon 100% fund expenditure.

Deliverables Provided Six Months After Contract Award (August 1, 2015):

- Baseline techniques to identify forensic traces specific to a camera's make and model.
- Algorithm to determine the make and model of an image's source device using basic fusion these forensic traces.
- Single-core software implementations of these algorithms using C/C++ and/or MATLAB, along with testing and validation performed on approximately 20-25 cameras models.

Deliverable Provided Twelve Months After Contract Award (February 1, 2016):

- Improved techniques to identify forensic traces specific to a camera's make and model.
- Updated algorithm to determine the make and model of an image's source device using improved fusion these forensic traces.
- Software implementations of these algorithms using MATLAB, along with testing and validation performed on approximately 50 cameras models.
- Parallel design and implementation of algorithms developed as six month deliverables in C/OpenMP that are suitable for use on multi-core CPUs.

- Baseline forensic algorithm to identify the imaging device type (i.e., camera, scanner, etc).
- Design of algorithm to gather, process, and ensure the integrity of crowd-sourced training data.

Deliverables Provided Eighteen Months After Contract Award (August 1, 2016):

- Enhanced techniques to identify forensic traces specific to a camera's make and model.
- Improved algorithm to determine the make and model of an image's source device using more sophisticated fusion of forensic traces.
- Incorporation of algorithms to forensically identify basic editing that may affect camera model identification (multiple JPEG compression, resizing, etc.)
- Software implementations of these algorithms using MATLAB, along with testing and validation performed on approximately 75 cameras models.
- OpenMP implementations for multi-core CPUs of algorithms developed as 12 month deliverables and the development of highly-parallel algorithms for forensics and data fusion.
- C/CUDA-based implementation of the above developed techniques for GPUs.
- Implementation of a crowd-sourced training algorithm, along with initial testing and validation using an image set gathered from Internet photo-sharing sources.

Time Line Chart by Task

The following table provides a breakdown of the project timeline, organized in terms of the various proposed tasks.

Task	Description	Phase 1	Phase 2	Phase 3
#1	Baseline multimedia forensic techniques to identify camera traces.	✓		
	Improved multimedia forensic techniques to identify camera traces.		✓	✓
	Algorithm development to forensically identify editing that can be used as anti-forensic countermeasures.			✓
	Testing and validation.	✓	✓	✓
#2	Baseline data fusion algorithm to perform device model identification.	✓		
	Improved data fusion algorithm to perform device model identification.		✓	✓
	Incorporation of traces of anti-forensic editing into fusion algorithms.			✓
	Testing and validation.	✓	✓	✓
#3	Single core implementations of forensic processing workflows.	✓	✓	✓
	GPU and Multi-Core accelerated Forensic Processing Workflows.		✓	✓
	Software testing and verification.	✓	✓	✓
#4	Crowd-sourced techniques for image data acquisition for the forensic algorithms and software training.		✓	✓

Table 1: Project schedule and milestones.

1.2 What was accomplished under these goals?

Summary

Below is a summary of the major accomplishments achieved under this project.

- We developed two new algorithms to perform camera model identification using demosaicing traces. Both algorithms are described below.

- One algorithm was designed to be very computationally efficient. This algorithm is described in Section 1.2.2.
  - This algorithm operates by obtaining a least-squares estimate of a camera's demosaicing filter, then uses these filter estimates a camera model identification traces.
  - Experimental results show that this algorithm can can identify the make and model of an image's source camera with an **average accuracy of 86.81%**.

- One algorithm was designed to be highly accurate and yield reliable results in re-compressed or potentially post-processed images. This algorithm is described in Section 1.2.3.
  - This algorithm operates by using an advanced machine learning algorithm to search for statistical traces in features we call 'demosaicing residuals'.
  - Experimental results show that this algorithm can can identify the make and model of an image's source camera with an **average accuracy of 99.65%**.

- We implemented an algorithm to perform camera model identification using information stored in an image's header file such as JPEG quantization tables, Huffman tables, image height and width. This algorithm is described in Section 1.2.4.
  - This algorithm is capable of identifying a small set of possible source camera models without making classification errors.
  - This algorithm is designed to be integrated into our hierarchical decision fusion framework.

- We designed a new hierarchical decision fusion framework to combine multiple traces in order to perform highly accurate and computationally efficient camera model identification.
  - This framework integrates each of the algorithms described above.
  - Experimental results show that when the computationally efficient demosaicing trace based algorithm is integrated into our framework, it can perform camera model identification with an **average accuracy of 99.96%**. This corresponds to a **13.15 percentage point increase in accuracy**.

- We developed a highly parallel GPU implementation of our demosaicing residual based algorithm that resulted in an **over one order of magintude decrease in runtime**.

- We collected a large scale image database for testing and validation purposes. Our database contains approximately 35,160 images collected using 71 different camera models. Additionally, we created a software application to crawl the photo sharing website Fickr and gather public domain images for the creation of a very large scale database.

- We created **a software package titled the *Source Camera Model Identification Tool*** containing implementations of all algorithms developed under this project. A description of this software module is give in Section 2.3 of the "Products" section of this report.
  - This software package can be run on individual images or large data sets from either a graphical user interface or a Matlab command line.
  - We created a 32 page user manual describing how to install our software, use it to identify the source camera model of images under investigation, and how to train it to recognize new camera models.

1.2.1 Background Information

The set of physical components and processing algorithms that a device uses to capture a digital image is known as its image processing pipeline. The image processing pipeline of a typical digital camera is shown in Fig. 1.

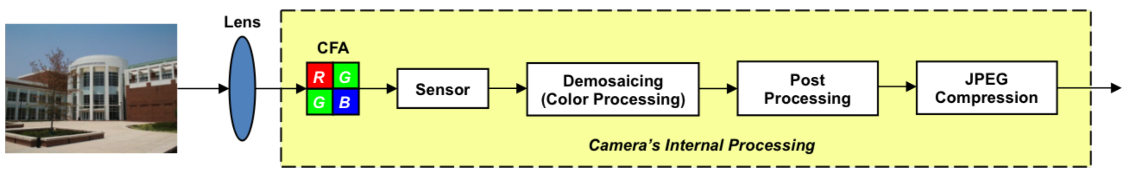

Fig. 1: The image processing pipeline of a digital camera

Light enters a camera by first passing through a lens. Since most imaging sensors are only capable of measuring one color of light at each pixel location, the light next passes through a color filter array (CFA), which is an optical filter that allows only one color of light (red, green, or blue) to hit the imaging sensor at a particular pixel location. The sensor then measures the light intensity of the corresponding color band at each pixel location, and produces an image constructed of three partially sampled color layers. Next, the unobserved color values at each pixel location are interpolated using a process known as demosaicing. Demosaicing algorithms are typically proprietary and are updated by a camera manufacturer with each new model. After this, the image may undergo post-processing, such as white balancing and color correction, followed by JPEG compression before the final image file is output or stored.

While the image processing pipelines of different devices are generally composed of common elements (e.g. lens, sensor, color processing algorithms, image compression, etc.), the specific implementation and characteristics of each element typically varies from manufacturer-to-manufacturer and from model-to-model. Additionally, different types of imaging devices, such as digital cameras and scanners, may use image processing pipelines composed of slightly different sets of elements. The image processing pipeline is typically consistent, however, across all devices that share a common model and manufacturer.

1.2.2 Computationally Efficient Camera Model Identification Algorithm Using Demosaicing Filter Estimates

This algorithm identifies the make and model of an image's source camera using traces left in the image by the camera's internal demosaicing algorithm. It is designed to be computationally efficient, and to allow investigators to balance the trade-off between camera model identification accuracy and computational complexity (i.e. processing time). The development and implementation of this algorithm addressed Tasks 1, 2, and 3 of this project.

Algorithm Overview

This algorithm builds upon previous work by Swaminathan et al.[1] in which the demosaicing process inside a camera is modeled as a set of linear shift-invariant filter. To compensate for the fact that the vast majority of demosaicing algorithms used by modern digital cameras are nonlinear and adaptive, this algorithm divides each color channel (red, green, and blue) of an image into separate gradient/texture regions (horizontal gradients, vertical gradients, and smooth) where the demosaicing process can locally be approximated as linear. It then obtains a different estimate of the demosaicing filter for each pairing of color channel and gradient region. These filter estimates are then grouped together to form a single camera model identification feature set. A support vector machine (SVM) is then trained to perform camera model identification on the basis of this feature set.

The original algorithm proposed by Swaminathan et al. has a significant shortcoming; it is too computationally expensive to use all pixels in an image to perform demosaicing filter estimation. To mitigate this, Swaminathan et al. proposed a heuristic for selecting a small window of an image to use for performing filter estimation and camera model identification. Our initial experiments revealed, however, that their heuristic can yeild inconsistent and highly variable results. Furthermore, it cannot precisely control the trade-off between computational complexity and accuracy, i.e. sometimes small image windows will still require significant processing and sometimes large image windows will not yeild sufficiently accurate filter estimates to perform camera model identification.

In this project, we developed a new algorithm that is able to search through an entire image and find the set of the N best pixels throughout the entire image to use for performing demosaicing filter estimation. To accomplish this, we first derived a lower bound on the Frobenius norm of demosaicing filter's estimation error covariance matrix. Insights gained from this lower bound allowed us to identify that better filter estimates can be produced by using the N pixels whose local neighborhoods had the highest variance to perform estimation. Furthermore, we developed a quick way to identify an approximation of this set of pixels and to ensure that they have proper gradient orientation. We do this by using the set of pixels along edges in an image with the N largest edge strengths.

Using this information, we devised a method to allow an investigator to balance the trade-off between computational complexity and camera model identification accuracy when performing demosaicing trace estimation as described above. Furthermore, at a fixed computational cost, our algorithm results in a significantly higher camera model identification accuracy than Swaminathan et al.'s approach. Similarly, our algorithm significantly reduces the computational cost needed to achieve a particular minimum camera model identification accuracy.

Additionally, we investigated two techniques to report confidence scores along with our camera model identification results. Instead of using a traditional SVM to perform classification, we used a P-SVM (i.e. an SVM with Platt scaling) as well as a neural network to return camera model

[1] A. Swaminathan, M. Wu, and K. J. R. Liu. "Nonintrusive component forensics of visual sensors using output images." *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, Mar. 2007, pp. 91-106.

identification confidence scores and decisions. An experimental evaluation suggests that the P-SVM approach yields more reliable results.

Testing and Validation Results

We performed two sets of experiments to test and validate the performance of this algorithm. In the first experiment, we evaluated our algorithm's ability to perform large-scale camera model identification on a set of images from 71 different camera models. To conduct this experiment, we used approximately 300 images from each camera model in our database for a total of 20,945 images. See Section 1.2.7 for details of this database. Our classifier was trained and tested using five-fold cross-validation. Specifically, the data was divided into five approximately equal folds. Four data folds were used to train the classifier, then the trained classifier was used to determine the source images in the remaining fold. This process was repeated five times, each time using a different set of training and testing folds, and the overall classification accuracy was averaged over all five experiments.

Results of this experiment show that our computationally efficient algorithm was **able to correctly identify the model of an image's source with 86.81% accuracy**. These results show that our algorithm can be used to accurately identify the model of an image's source camera on large sets of data. Later in this report, we will show that when this algorithm is incorporated into our camera model trace data fusion framework, we can achieve a camera model identification accuracy of 99.96%.

In our second set of experiments, we characterized the our algorithm's trade-off between computational complexity (i.e. runtime) and identification accuracy, and showed that it yields a significant performance gain over the approach proposed by Swaminathan et al. To conduct this experiment, we used a set of images 1300 images captured using 13 different camera models from our experimental database. These models consisted of 9 cell phone cameras, 2 point-and-shoot cameras, and 2 digital SLRs. We then fixed a computational cost (as measured by the size of the data matrix used when performing least-squares demosaicing filter estimation) and measured the camera model identification accuracy of both our approach and Swaminathan et al.'s approach using five-fold cross validation.

Results of this experiment are shown in Fig. 2. From this experiment, we can see that our algorithm results in a significantly greater classification accuracy at a fixed computational cost. For example, for a data matrix length of n = 2515 per gradient region, our method achieves a classification accuracy of 71.3% accuracy. By contrast, Swaminathan et al.s method achieves a classification accuracy 47.3%. This 24.0% gain in classification accuracy is marked using the green double arrow. Similarly, Fig. 2 also shows that a given classification accuracy can be achieved by our algorithm at a much lower computational cost. The decrease in computational cost achieved using our proposed method is marked using the red double arrow.

1.2.3 Advanced Camera Model Identification Algorithm Using Demosaicing Residuals

We developed an advanced, machine learning based algorithm that identifies the make and model of an image's source camera using traces left in the image by the camera's internal demosaicing algorithm. This algorithm is designed to be highly accurate and robust to post-processing. The development and implementation of this algorithm addressed Tasks 1, 2, and 3 of this project.

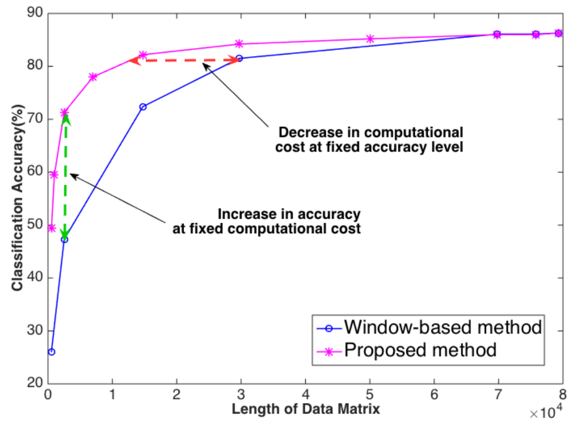

Fig. 2: Tradeoff between computational complexity vs. classification accuracy for our method (labeled as 'proposed method') and Swaminathan et al.'s method (labeled as 'window-based method'). The length of the data matrix used when performing least-squares estimation is used as a measure of computational complexity.

Algorithm Overview

This algorithm is a powerful, machine learning-based algorithm designed to gather traces left in an image by its source camera's demosaicing algorithm without using explicit parametric models. Instead, it searches for content-independent pixel value and color channel relationships introduced by the demosaicing process. To accomplish this, the color values of an image are first resampled according to a fixed color filter array (CFA) pattern. Next, the image is re-demosaiced using a variety of both linear and nonlinear demosaicing algorithms. After this, a set of *demosaicing residuals* are calculated by subtracting each re-demosaiced image from the original image. This suppresses an image's scene contents and allows us to search for traces left by the camera's demosaicing algorithm. After this, a set of third-order co-occurrence matrix features are computed from the demosaicing residuals using the co-occurrence patterns shown in Figs. 3 and 4. These co-occurrence matrices are multi-dimensional histogram approximations of the joint distribution of demosaicing residuals that are used to expose unique intra-channel and inter-channel correlations introduced by a camera model's demosaicing algorithm. Finally, the co-occurrence matrices are pooled together and used as a high-dimensional camera model identification feature set.

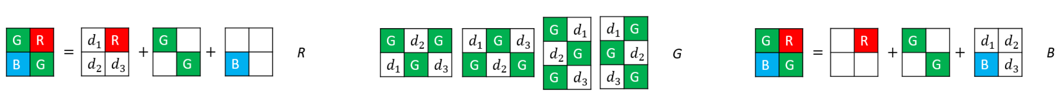

Fig. 3: Intra-channel co-occurrence patterns for red (left), green (middle) and blue (right) channels.

Camera model identification is performed using a powerful, specially designed ensemble classifier. This classifier is built from a set of binary classifiers trained to distinguish between each pair of potential camera models. A multi-class classifier is created from these intermediate binary classifiers using the all-against-all strategy, and decisions of these intermediate binary classifiers are fused using a majority voting protocol. Each binary classifier is itself an ensemble classifier made of a large set of Fischer linear discriminant classifiers that pull a random subset of features from the full co-occurrence feature set. An overview of our entire feature extraction and classification framework's architecture used under this algorithm is shown in Fig. 5.

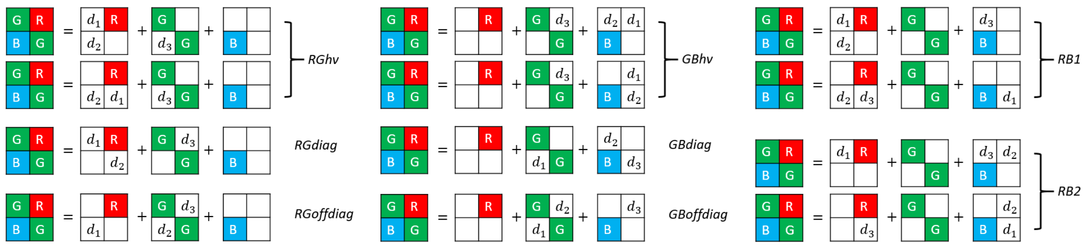

Fig. 4: Inter-channel co-occurrence patterns between red and green (left), green and blue (middle) and red and blue (right) channels.

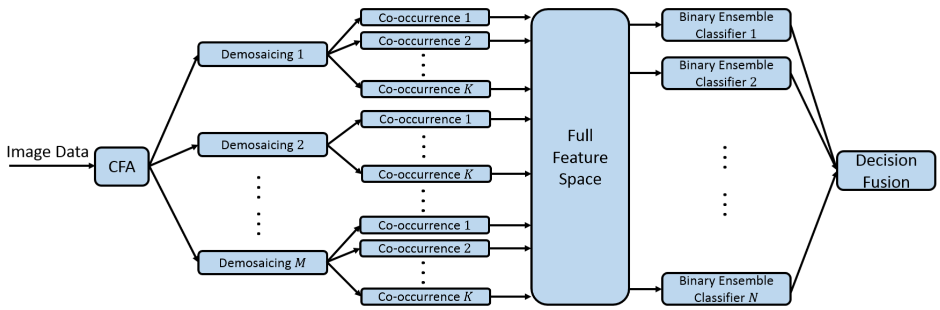

Fig. 5: Architecture of our demosaicing residual based camera model identification framework.

Testing and Validation Results

We performed several sets of experiments to test and validate the performance of this algorithm. In our first experiment, we evaluated our algorithm's ability to perform large-scale camera model identification on a set of images from 65 different camera models. To conduct this experiment, we used 33,568 images from our database described in Section 1.2.7. Each image was then divided into 512×512 pixel blocks, and blocks with insufficient lighting or texture were excluded, resulting in a set of 606,424 distinct image blocks. We then trained our classifier using approximately 90% of these blocks and used our algorithm to identify the make and model of the source camera of the remaining 10% of blocks.

Using this advanced algorithm, we achieved an **average camera model classification accuracy of 99.65%**. These results are the highest reported camera model classification accuracy of any existing algorithm. Furthermore, we note that to the best of our knowledge, these results have been achieved on a database containing more camera models than any other algorithm. Recent publications have referred to an intermediate version of our algorithm published in 2015 as the state-of-the-art algorithm for performing camera model identification[2].

Additionally, we performed a series of experiments to determine the robustness of this algorithm to image post-processing operations. To conduct these experiments, we used images from the 12 different camera models shown in Table 2. We then post-processed these images by re-compressing them using JPEG quality factors of 90 and 70, as well as performing contrast enhancement using gamma correction with $\gamma = 0.8$. These operations were selected because re-compression and color correction are processing operations commonly applied to images by online photo sharing sites and social media applications. After this, we trained our classifier using 90%

[2]L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E. Delp, and S. Tubaro, "First Steps Towards Camera Model Identification with Convolutional Neural Networks." *IEEE Signal Processing Letters*, Dec. 2016.

of each data set and used it to determine the source of the remaining 10% of post-processed image.

Camera Number	Make	Model
1	Canon	PC1234
2	Apple	iPhone 4s
3	Apple	iPhone 5
4	Apple	iPhone 5s
5	Apple	iPhone 6
6	Motorola	Moto X
7	Nikon	D7100
8	Nokia	Lumia 920
9	Samsung	Galaxy S4
10	Samsung	Galaxy S5
11	Sony	A6000
12	Sony	NEX-5TL

Table 2: Camera models used in our robustness evaluation experiments.

Our advanced algorithm was able to correctly identify the source of gamma corrected images with an average accuracy of 99.65%. Similarly, it was able to correctly identify the source of JPEG images with an average accuracy of 99.64% for images recompressed using a quality factor of 90 and with an average accuracy of 98.89% for images recompressed using a quality factor of 70. Tables 3, 4, and 5 contain confusion matrics showing detailed results of these experiments with individual correct identification accuracies listed in bold. The results of these experiments indicate that our advanced algorithm is highly robust to several post-processing operations.

		True Model											
		1	2	3	4	5	6	7	8	9	10	11	12
Identified Model	1	**100.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.000	**99.900**	0.100	0.000	0.112	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	3	0.000	0.000	**99.800**	0.500	0.112	0.100	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.000	0.000	0.000	**99.400**	2.354	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.000	**97.422**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	6	0.000	0.100	0.000	0.000	0.000	**99.800**	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000	0.000	0.000	0.000
	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000	0.000	0.000
	9	0.000	0.000	0.000	0.000	0.000	0.100	0.000	0.000	**100.000**	0.000	0.000	0.000
	10	0.000	0.000	0.100	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000
	11	0.000	0.000	0.000	0.100	0.000	0.000	0.000	0.000	0.000	0.000	**99.390**	0.000
	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.610	**100.000**

Table 3: Confusion matrix for camera model identification performed on JPEG compressed images using a JPEG quality factor = 90.

Resiliance to Anti-Forensic Camera Model Falsification Attacks

An intelligent information attacker can modify an image by using anti-forensic algorithms in an attempt to disguise its true source. We developed a protocol to train our algorithm to both detect anti-forensic attacks and identify the model of the camera that actually captured an attacked image.

		True Model											
		1	2	3	4	5	6	7	8	9	10	11	12
Identified Model	1	**100.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.000	**99.700**	0.100	0.200	0.000	0.000	0.100	0.000	0.000	0.000	0.000	0.000
	3	0.000	0.100	**98.600**	0.700	0.448	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.000	0.100	0.900	**96.200**	3.812	0.100	0.000	0.000	0.000	0.000	0.305	0.184
	5	0.000	0.000	0.100	2.400	**95.404**	0.000	0.000	0.255	0.000	0.100	0.000	0.000
	6	0.000	0.000	0.000	0.000	0.000	**99.200**	0.000	0.000	0.000	0.100	0.000	0.000
	7	0.000	0.000	0.300	0.100	0.000	0.000	**99.800**	0.000	0.000	0.000	0.000	0.000
	8	0.000	0.100	0.000	0.000	0.000	0.000	0.000	**99.745**	0.000	0.000	0.000	0.000
	9	0.000	0.000	0.000	0.300	0.000	0.100	0.000	0.000	**100.000**	0.000	0.000	0.000
	10	0.000	0.000	0.000	0.100	0.224	0.500	0.100	0.000	0.000	**99.800**	0.000	0.000
	11	0.000	0.000	0.000	0.000	0.112	0.100	0.000	0.000	0.000	0.000	**98.780**	0.368
	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.915	**99.449**

Table 4: Confusion matrix for camera model identification performed on JPEG compressed images using a JPEG quality factor = 70.

		True Model											
		1	2	3	4	5	6	7	8	9	10	11	12
Identified Model	1	**100.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	0.000	**99.900**	0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	3	0.000	0.100	**99.700**	0.300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.000	0.000	0.200	**99.400**	3.027	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.300	**96.973**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	6	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000	0.000	0.000	0.000
	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000	0.000	0.000
	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000	0.000
	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**100.000**	0.000	0.000
	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**99.848**	0.000
	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.152	**100.000**

Table 5: Confusion matrix for camera model identification performed on gamma corrected images with $\gamma = 0.8$.

Anti-Forensic Attack Description: We define the true camera model γ_A as the model of the camera that actually captured an image. Additionally, we define the target camera model γ_B as the model of the camera that an attacker wants investigators to believe was used to capture an image. A camera's demosaicing algorithm is modeled as linear interpolation implemented by convolving a partially color sampled image S output by the sensor with the camera's demosaicing filter θ to produce the final image $I = S * \theta$. The attack proceeds along the following steps:

1. Obtain a least-squares estimate the target camera's demosaicing filter θ_B using either our algorithm described in Section 1.2.2, the algorithm proposed by Swaminathan et al.[3], or a similar algorithm.

2. Pass the image under attack I through a "synthetic" color filter array (CFA) by resampling I such that only color values directly observed by the sensor are retained. This will reproduce S.

[3] A. Swaminathan, M. Wu, and K. J. R. Liu. "Nonintrusive component forensics of visual sensors using output images." *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, Mar. 2007, pp. 91-106.

3. Produce the anti-forensically attacked image $\tilde{I}$ by using the estimated demosaicing filter from the target camera to re-demosaic S such that $\tilde{I} = S * \theta_B$.

After this anti-forensic attack, the attacked image $\tilde{I}$ will have demosaicing traces associated with camera γ_B even though it was captured using camera γ_A.

Anti-Forensic Attack Validation: We performed experimental validation and performance evaluation of the anti-forensic attack described above. Experiments were performed by capturing 300 unaltered images from each of the cameras listed below in Table 6.

Table 6: True camera models in our database

Model No.	True Camera Models
1	Canon PC1234
2	Canon Powershot G10
3	Samsung Galaxy S3
4	Sansung Galaxy S4
5	iPhone 4s
6	iPhone 6

The anti-forensic attack was then used to make each image appear as if it was captured using a different target camera model, resulting in a set of 10,800 images from 36 possible (true, target) camera model pairings. Swaminathan et al.'s camera model identification algorithm was then used to determine the source of each anti-forensically modified image.

The overall success rate of the attack (i.e. the target model was identified instead of the true model) was 97.34%. Detailed experimental results are shown below in Table 7.

True Model \ Target Model	1	2	3	4	5	6
1	$\star$	95.33	98.33	99.00	93.67	98.67
2	99.00	$\star$	98.33	99.00	93.33	98.67
3	99.00	95.67	$\star$	99.00	93.67	98.67
4	99.00	95.33	98.00	$\star$	93.67	98.67
5	99.00	95.67	98.33	99.00	$\star$	98.67
6	99.00	95.67	98.33	99.00	93.33	$\star$

Table 7: Successful attack rate for 30 true and target camera model parings ($\star$ denotes no attack is necessary since the true and target model are the same).

Anti-Forensic Attack Detection and True Camera Model Identification: Because this attack relies on a linear model of a camera's demosaicing filter, it is not able capture or falsify demosaicing traces left by nonlinear components of a camera's demosaicing algorithm. By contrast, our advanced algorithm that utilizes demosaicing residuals to perform camera model identification can observe these nonlinear traces.

To defend against the anti-forensic attack described above, we trained our demosaicing residual-based camera model identification algorithm using both authentic images and images whose source camera has been falsified. Each pairing of true and target camera model was labeled as a different class so that our classifier could identify the true source of an image (i.e. detect that the true and target camera model are different, and determine the true model).

We evaluated the performance of this approach on the experimental database consisting of images captured by cameras listed in Table 6. Experimental results show that **our approach was able to detect anti-forensically attacked images and determine their true origin with 99.81% accuracy.** Table 8 shows detailed results our approach's accuracy at detecting the true camera model of falsified images.

		Target Model					
		1	2	3	4	5	6
True Model	1	100.00	99.80	100.00	100.00	100.00	99.40
	2	100.00	100.00	99.81	99.54	100.00	98.93
	3	100.00	98.93	100.00	100.00	100.00	100.00
	4	100.00	100.00	98.71	100.00	99.58	100.00
	5	100.00	100.00	100.00	100.00	99.57	100.00
	6	100.00	99.82	99.19	100.00	100.00	100.00

Table 8: True source identification accuracies for anti-forensically falsified camera models.

1.2.4 JPEG Header Trace Extraction

We implemented an algorithm to extract forensically significant traces from an image's JPEG header. These traces are then used to eliminate camera models that do not produce images with an identical set of JPEG header traces and identify a small set of possible source camera models. This algorithm was designed to be implemented into our data fusion framework described in Section 1.2.5. The development and implementation of this algorithm addressed Tasks 1, 2, and 3 of this project.

<u>Algorithm Overview</u>

An image's JPEG header contains several forensically significant traces. In this project, we utilized discrete cosine transform (DCT) quantization tables, Huffman code tables, and the image's height and width (in pixels) as forensic traces. Since an JPEG compression has a significant impact on the quality of an image, most digital camera manufacturers design their own proprietary DCT quantization tables. Similarly, most camera manufacturers develop their own Huffman coding tables to use during the lossless portion of JPEG compression. Additionally most camera models are only capable of producing images in a small number of sizes.

While the values these traces are not typically unique to a particular camera model, very few camera models will produce an identical set of traces. We use these traces to sort camera models into groups called *equivalence classes*. All camera models in an equivalence class are capable of producing an identical set of JPEG header traces.

We developed a software module to read these traces from an image's JPEG header and compare them to a hash table of precomputed equivalence classes. It can be trained, i.e. used to generate a hash table of equivalence classes, by extracting traces from a set of images whose source is already known and matching camera models with identical traces. This module was written in C++ with a MEX interface into Matlab and is designed to be integrated into our camera model trace fusion framework described

1.2.5 Data Fusion Framework for Combining Camera Model Identification Traces

We developed a hierarchical data fusion framework to combine information gathered from an image's JPEG header with demosaicing filter traces. This framework significantly increases the overall camera model identification accuracy of our algorithms described above, particularly our algorithm described in Section 1.2.2. The development and implementation of this framework addressed Tasks 2, and 3 of this project.

Framework Overview

Our data fusion framework is designed to exploit intrinsic information hierarchies that naturally occur within camera model identification fingerprints. For example, JPEG header traces can be used to eliminate possible source camera models without running the risk of making decision errors, but they cannot be used to identify a single source camera model.

Our framework uses a tree-based decision structure that exploits these latent information hierarchies. Early stages of the framework eliminate camera models that cannot possibly be the image's source camera. Later stages identify a single source camera model from the reduced set of possible camera models. This divide-and-conquer approach reduces the number of camera models that later stages need to choose from, thus increasing the accuracy of classifiers used at these stages. Since these early stage decisions are error-free, this reduces the overall error rate of the entire framework.

An overview of of our decision fusion algorithm is described below:

1. JPEG header information such as quantization tables, image size, etc. is extracted from an image.
2. These traces are used to sort image into one of N equivalence classes. Each equivalence class is the set of all camera models able to produce images with the same JPEG header information.
3. Demosaicing filter traces are extracted from the image using one of the two algorithms described in Sections 1.2.2 and 1.2.3.
4. A trained classifier is used to differentiate between each camera model in the equivalence class on the basis of the demosaicing filter traces.

Testing and Validation Results

We performed an experiment to test and validate the performance of our data fusion framework. To do this, we used our algorithm's ability to perform large-scale camera model identification on a set of images from 71 different camera models. To conduct this experiment, we used approximately 300 images from each camera model in our database described in Section 1.2.7 for a total of 20,945 images. Demosaicing trace extraction and final camera model identification decisions were made using our computationally efficient algorithm described in in Section 1.2.2. Finally, our framework and the classifiers integrated into it were trained and tested using five-fold cross-validation.

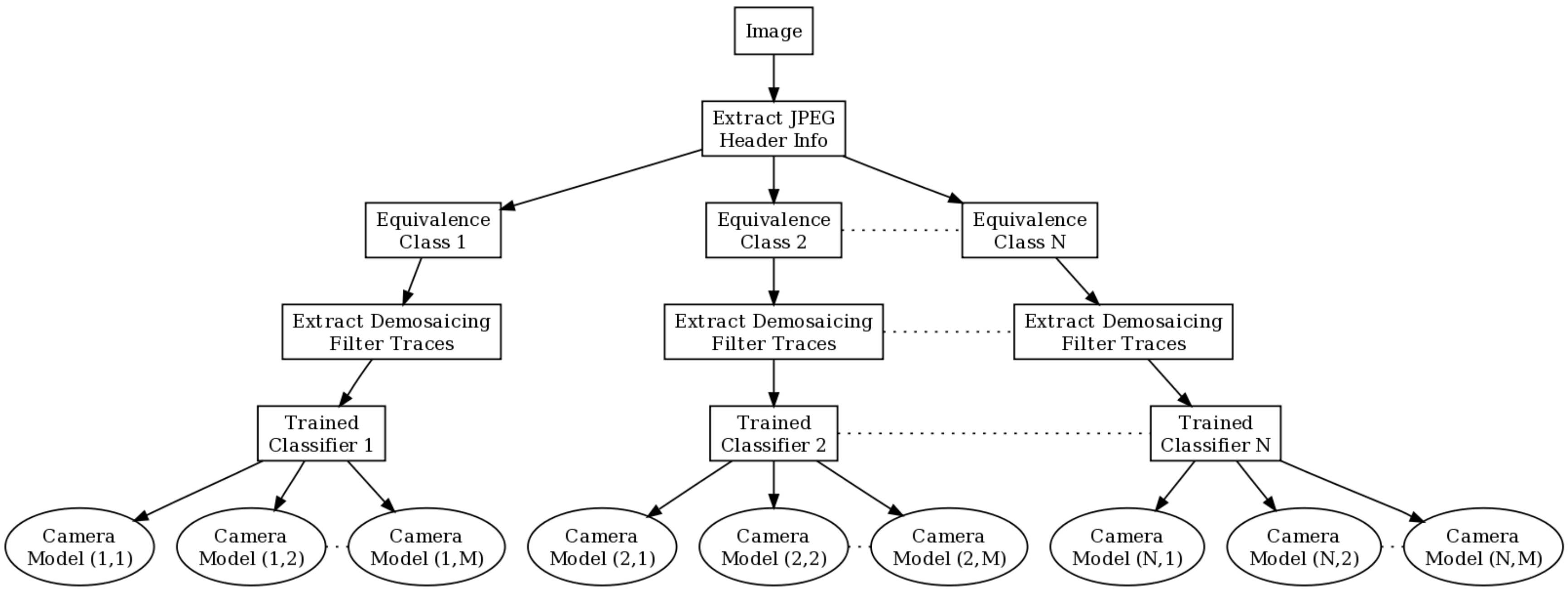

Fig. 6: Flowchart providing an overview of our camera model identification framework (including hierarchical fusion of traces).

Results of this experiment show that **our hierarchical data fusion framework was able to correctly identify the model of an image's source with 99.96% accuracy**. Without using this framework, our algorithm described in in Section 1.2.2 was only able to achieve a classification accuracy of 86.81%. These results show that **our data fusion framework was able to provide a 13.15 percentage point increase in accuracy**.

1.2.6 Design and GPU Implementation of Parallel Algorithms

The camera model identification that utilizes demosaicing residual described in Section 1.2.3 is very computationally expensive. To mitigate this issue, we developed a parallel version of this algorithm that is suitable for implementation on graphics processing units (GPUs). Our C/CUDA implementation of this algorithm reduces the runtime of performing camera model identification by over an order of magnitude. The development and implementation of this algorithm addressed Tasks 1 and 3 of this project.

Algorithm Overview

The most computationally expensive element of our algorithm described in Section 1.2.3 is the calculation of co-occurrence matrices from demosaicing residuals. To reduce this computational cost, our algorithm exploits data level parallelism when building co-occurrence histograms, i.e. the residual data is distributed across many GPU cores, intermediate calculations are performed, then the resulting data is combined to form the co-occurrence histograms. To further improve efficiency, we developed a method to process multiple images per call, which avoids having to pay the costs of re-initializing the GPU in between every image. This was improved by developing a technique to prevent GPU idling by splitting process into two streams. This allows the CPU to fetch the next image from the hard drive while the GPU kernel is executing.

Additionally, we created a technique to reduce the data transfer bottleneck and improve GPU workload efficiency by processing multiple co-occurrence patterns simultaneously per image. This resulted in a significant speedup over processing one co-occurrence pattern at a time.

Testing and Validation Results

To measure the computational efficiency gains achieved by our parallel algorithm we performed a series of experiments. Our algorithm was run on an Nvidia GTX 980 GPU with 4 GB of onboard RAM, along with a CPU version run on a computer with a 3.4 GHz Intel i7-4770 processor and 16 GB of RAM. Testing and validation of GPU implementation of parallel algorithm performed on 1,000 image patches of size ranging from 128×128 pixels to 1024×1024 pixels.

In our first experiment, we performed a runtime comparison between our parallel algorithm capable of running on a GPU and our original sequential algorithm running on a CPU. Experimental results are shown below in Figure 7. **Our improved parallel algorithm achieved an order of magnitude speed-up when compared to our original sequential algorithm**.

Additionally, we performed a runtime comparison between our individual algorithm calls for calculating co-occurrence matrices for six different co-occurrence patterns and our improved technique using a single algorithm call capable of processing all six co-occurrence patterns at once. Experimental results are shown below in Figure 8. **Our improved technique to process multiple co-occurrence patterns in parallel achieved an addtional speed-up of greater than $2\times$**.

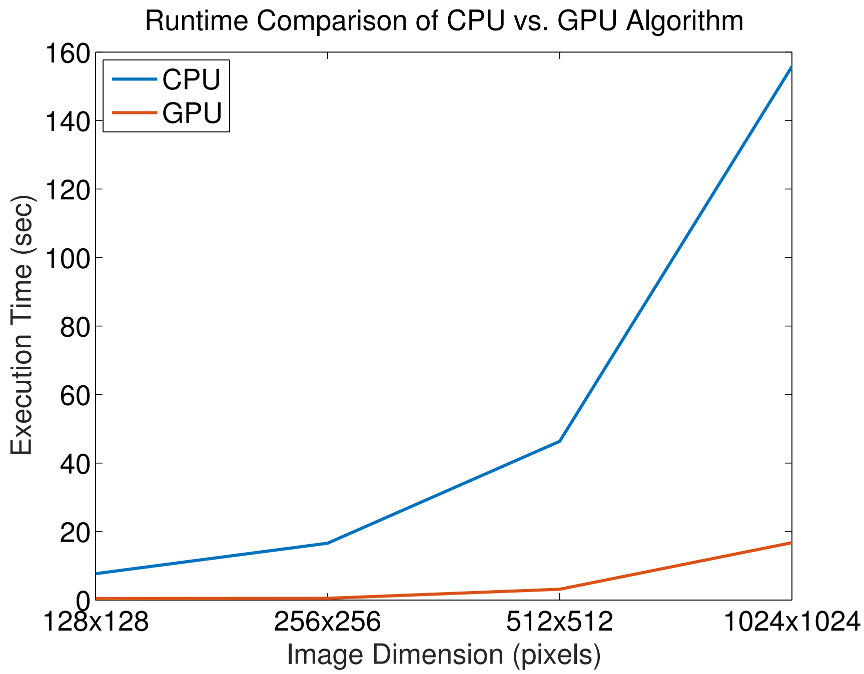

Fig. 7: Runtime comparison for calculating co-occurrence matrices for 1,000 images using our improved parallel algorithm implemented on a GPU and our original sequential algorithm implemented on a CPU.

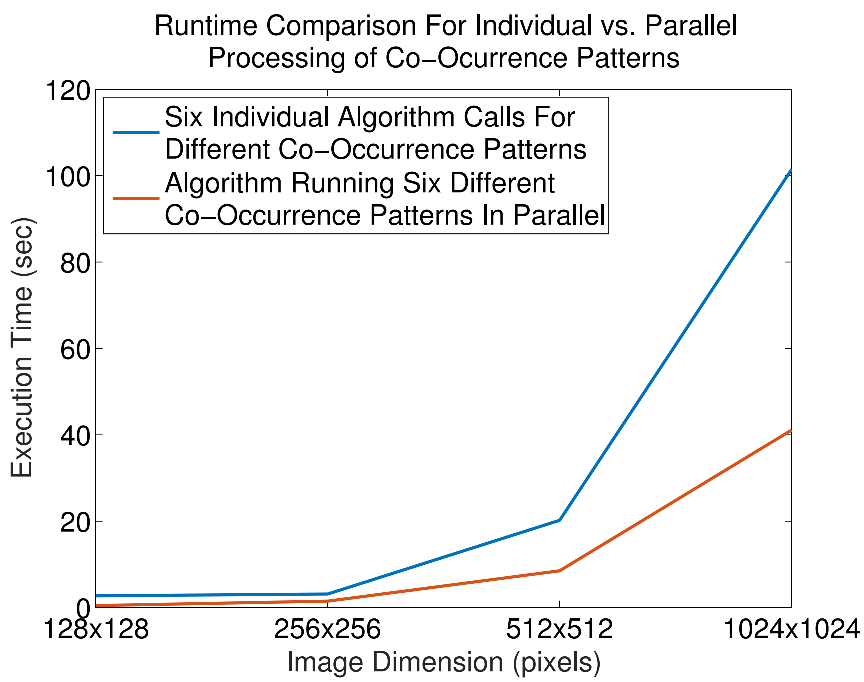

Fig. 8: Runtime comparison for calculating co-occurrence matrices using six different co-occurrence patters for 1,000 images using individual calls of our algorithm and our improved technique capable of processing multiple co-occurrence patterns at once.

1.2.7 Large Scale Data Collection

A large scale image database was collected in order to perform testing and validation of the camera model identification algorithms and software tools developed under this award. The collection of this database addressed Task 4 of this project.

Our database contains approximately 35,160 images collected using 71 different camera models, including cell phone cameras, digital SLRs, and point-and-shoot cameras. All images were manually collected to ensure that their provenance could accurately be recorded and to ensure that they underwent no post-processing or enhancement. At least 300 images were collected using each camera model in the database. For several camera models, multiple devices (of the same make and model) were used to collect images. Images were captured using a variety of scenes (both indoor and outdoor) and lighting conditions. All images were captured and stored as JPEGs using the camera's default settings.

Additionally, a software tool written in JAVA was created to crawl the photo sharing website Flickr[4], and download public domain images in order to create a large-scale database for testing and validation purposes. This was done to enable future efforts to perform very large scale training and validation of our camera model identification framework.

1.3 Opportunities for training and professional development

This project afforded several opportunities for training and professional development. A total of five graduate students (four doctoral and one masters student) were involved in developing the algorithms and software created under this project. As a result, this project enabled these students to receive advanced training in multimedia forensics, signal processing, and machine learning research from PI Stamm with additional training from Co-PI Kandasamy.

Additionally, this project facilitated the professional development of both PI Stamm and the graduate students involved in this project by enabling attendance at major research conferences. Both PI Stamm and graduate students involved with this project were able to attend and present research conducted under this project at the 2015 IEEE International Workshop on Information Forensics and Security (WIFS) and the 2016 IEEE International Conference on Image Processing (ICIP).

1.4 Dissemination of results to communities of interest

This project led to the publication of two papers listed in Section 2.1. Talks associated with these papers were given at the 2015 IEEE International Workshop on Information Forensics and Security (WIFS) and the 2016 IEEE International Conference on Image Processing (ICIP). Furthermore, two journal papers describing research developed under this project are being prepared for submission to IEEE Transactions on Information Forensics and Security. These papers are also listed in Section 2.1.

In addition to these publication efforts, descriptions of the algorithms developed under this project, software demonstrations, and training sessions were provided to stakeholders during bi-annual performance review and project closeout meetings held in August 2015, February 2016,

[4] https://www.flickr.com/

and August 2016. Representatives from the Defense Forensics Science Center, the National Media Exploitation Center, the Digital Cyber Crime Center, the Federal Bureau of Investigation, the Department of Homeland Security, the Defense Advanced Research Projects Agency, and several other federal agencies attended these meetings

2 Products

2.1 Publications, conference papers, and presentations

The following peer-reviewed publications were produced as a result of work performed under this award:

C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features" *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, Rome, Italy, Nov. 2015, pp. 1-6.

Status: Published
Acknowledgment of Federal Support: Yes

X. Zhao and M. C. Stamm, "Computationally efficient demosaicing filter estimation for camera model identification" *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, Sep. 2016, pp. 151-155.

Status: Published
Acknowledgment of Federal Support: Yes

Additionally, the following publications are being prepared for submissions to peer-reviewed scientific journals as a result of work performed under this award:

C. Chen and M. C. Stamm, "Robust camera model identification using demosaicing residuals" *in preparation for submission to IEEE Transactions on Information Forensics and Security*, to be submitted Feb. 2017.

Status: In preparation for submission to IEEE Transactions on Information Forensics and Security
Acknowledgment of Federal Support: Yes (Will occur when paper is submitted)

X. Zhao and M. C. Stamm, "Efficient and scalable camera model identification using hierarchical feature fusion algorithms for big data environments." *in preparation for submission to IEEE Transactions on Information Forensics and Security*, to be submitted Mar. 2017.

Status: In preparation for submission to IEEE Transactions on Information Forensics and Security
Acknowledgment of Federal Support: Yes (Will occur when paper is submitted)

2.2 Databases

An image database was collected in order to perform testing and validation of the camera model identification algorithms and software tools developed under this award. This database contains approximately 35,160 images collected using 71 different camera models, including cell phone cameras, digital SLRs, and point-and-shoot cameras. At least 300 images were collected using each camera model in the database. For several camera models, multiple devices (of the same make and model) were used to collect images. Images were captured using a variety of scenes (both indoor and outdoor) and lighting conditions. All images were captured and stored as JPEGs using the camera's default settings.

Copies of this database were delivered to representatives from the Defense Forensics Science Center and the Science (DFSC) and Technology Integration Laboratory (STIL) at the closeout meeting for this project that occurred on August 25, 2016. We note that this image database is signficantly larger than the *Dresden Image Database*[5], which is currently the largest publicly available database used to test and validate multimedia forensic algorithms.

2.3 Technologies or techniques

This award led to the development of a software package titled the *Source Camera Model Identification Tool*. Screenshots of this tool are shown below in Fig. 9.

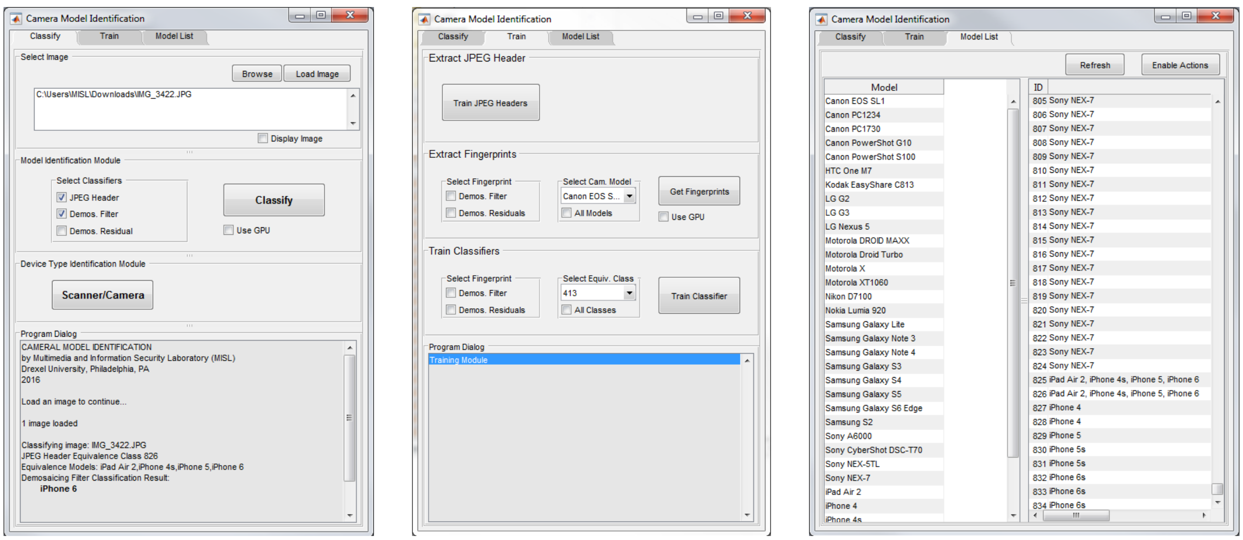

Fig. 9: Screen capture of our Matlab camera model identification software framework's graphical user interface while performing classification and while performing training.

The *Source Camera Model Identification Tool* software package includes implementations of all algorithms developed under this project, including separate modules for performing camera model identification (with the option to perform identification using all algorithms developed under this project both with and without feature fusion), large-scale feature extraction, and classifier training. It can be run using either a Graphical User Interface or from the command line in Matlab. Source code was written was written in Matlab, C/C++, and CUDA, and is provided along with the full software package. Furthermore, the software package includes all trained machine learning algorithms and classifiers it calls upon.

A 32 page user manual was created for the *Source Camera Model Identification Tool* to provide documentation and user instructions. The user manual includes directions on how to install the software package, perform camera model identification, feature extraction, and classifier training.

[5]The *Dresden Image Database* is available at `http://forensics.inf.tu-dresden.de/ddimgdb/`

Descriptions of all functions and subroutines invoked by this software package are included in the user manual. The user manual also includes an explicit funding acknowledgement along with a reference to the award number associated with this project.

Additionally, a software tool written in JAVA was created to crawl the photo sharing website Flickr[6], and download public domain images in order to create a large-scale database for testing and validation purposes.

Copies of this software package, including all documentation, source code, trained classifiers, and relevant executable files, along with copies of the JAVA-based tool designed to download large-scale data from Flickr, were delivered to representatives from the Science Technology Integration Laboratory (STIL) at the closeout meeting for this project that occurred on August 25, 2016.

[6] `https://www.flickr.com/`

3 Participants & Other Collaborating Organizations

3.1 Participants

The following individuals participated in this project and were supported through funding provided by this award (Award Number W911NF-15-2-0013).

- Matthew C. Stamm, Ph.D. (PI)
  - Role: Principal Investigator
  - Contribution to Project: Project management, algorithm development
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

- Nagarajan Kandasamy, Ph.D. (Co-PI)
  - Role: Co-Principal Investigator
  - Contribution to Project: Project management, algorithm development
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

- Xinwei Zhao
  - Role: Research Assistant
  - Contribution to Project: Programming/algorithm implementation, algorithm development, image training data acquisition
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

- Belhassen Bayar
  - Role: Research Assistant
  - Contribution to Project: Programming/algorithm implementation, algorithm development, image training data acquisition
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

- Chen Chen
  - Role: Research Assistant
  - Contribution to Project: Algorithm development, image training data acquisition
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

- Leland Machen
  - Role: Research Assistant
  - Contribution to Project: Programming/algorithm implementation, algorithm development
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

- Owen Mayer
  - Role: Research Assistant
  - Contribution to Project: Programming/algorithm implementation, graphical user interface development, image training data acquisition
  - Collaborated With Individual in Foreign Country: No
  - Traveled to Foreign Country: No

3.2 Other Collaborating Organizations

Nothing to report.

4 Impact

4.1 What is the impact on the development of the principal discipline of this project?

This project led to the development of several new forensic algorithms designed to determine which camera model captured an image. Specifically, two new camera identification algorithms and a new data fusion framework to combine forensic traces were developed under this project. This has led to two peer reviewed scientific papers that have already been published as well as two papers that are in preparation for submission to peer reviewed scientific journals. These publications are listed in Section 2.1.

We note that the algorithms developed under this project have significantly advanced efforts to forensically determine an image's source. In particular, the algorithm discussed in Section 1.2.3 currently produces the highest camera model identification accuracy reported in scientific literature to the best of our knowledge. Recent peer reviewed publications have referred to this algorithm as the state-of-the-art camera model identification algorithm[7].

4.2 What is the impact on other disciplines?

Nothing to report.

4.3 What is the impact on physical, institutional, and information resources that form infrastructure?

This project led to the development of the image database described in Sections 1.2.7 and 2.2. This database will facilitate significant future research towards the development of new and improved forensic algorithms designed to determine the source, authenticity, and processing history of digital images.

4.4 What is the impact on society beyond science and technology?

The algorithms developed under this project can be used by the community at large to aid in verifying the source and authenticity of digital images. In particular, these algorithms may be useful to news reporting agencies that wish to verify the source of their images, law enforcement agencies who wish to identify the source of images (particularly those involved in child exploitation cases), legal experts who wish to verify evidence used in criminal and civil proceedings, and military and defense organizations who wish to determine the origin and authenticity of signal intelligence.

[7] L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E. Delp, and S. Tubaro, "First Steps Towards Camera Model Identification with Convolutional Neural Networks." *IEEE Signal Processing Letters*, Dec. 2016.

4.5 What dollar amount of the award's budget is being spent in foreign countries?

None of this project's funding was spent in foreign countries. Travel to attend the IEEE International Workshop on Information Forensics and Security (WIFS) in Rome, Italy was paid for using funds provided to PI Stamm by Drexel University.

5 Budgetary Information

Amount funded to date: $374,940.00
Amount invoiced to ONR Regional office: $374,843.46 (Direct: $275,797.89, Indirect $99,045.57)
Remaining balance: $66.54

The amount invoiced includes salary/stipend for Belhassen Bayar, Xinwei Zhao, Owen Mayer, Chen Chen, and Leland Machen, Nagarajan Kandasamy, and Matthew Stamm. Additionally, it includes the purchase of cameras and other equipment necessary to collect, store, and process data for this project as well as travel to bi-annual progress meetings and the final closeout meeting.